Multimodal Visual Question Answering with Amazon Berkeley Objects Dataset

Aditya A V MT2024009
Gangasagar HL MT2024048
M B Ashish MT2024085

May 18, 2025

Abstract

Visual Question Answering (VQA) is a challenging multimodal task that requires a models to generate natural language responses to open ended questions based on a given image. In this work, we constrain the output space to one-word answers, focusing on the open ended setting using the Amazon-Berkeley Objects dataset. The project follows a structured pipeline comprising of automated data curation, baseline benchmarking using MultiModal VQA models namely BLIP-Base and Qwen-VL2—and fine-tuning via Low Rank Adaptation (LoRA) to enhance model performance within constrained resource settings. For evaluation, we employ multiple metrics including Exact Match (EM), BAAI score, and BERTScore to capture both lexical and semantic alignment with ground-truth answers. Our fine-tuned model demonstrates a significant performance improvement of 28.6% in Exact Match , 0.10 in BAAI Score and 0.04 in BERT(F1) Score over the best baseline.

1 Introduction

Visual Question Answering (VQA) is a multimodal task at the intersection of computer vision and natural language processing, where a system is required to answer open-ended questions about a given image using natural language. The goal is to develop models capable of jointly understanding visual content and linguistic queries, simulating a real-world interaction where users can ask arbitrary questions about visual scenes and expect coherent, contextually grounded responses.

The relevance of VQA extends beyond academic interest; it has practical implications for assistive technologies, human-computer interaction, and autonomous systems, where dynamic scene understanding and question-answering capabilities are essential. Despite significant advances, VQA remains a challenging task due to ambiguities in language, the need for fine-grained visual recognition, and the complexity of aligning visual and textual features.

In this work, we focus on a constrained version of the VQA task—producing one-word answers in an open-ended setting. We utilize the Amazon-Berkeley Objects dataset and benchmark performance using modern multimodal architectures. Our approach includes baseline evaluation with BLIP-Base and Qwen-VL2, followed by fine-tuning using Low-Rank Adaptation (LoRA) to enhance performance. Evaluation is conducted using metrics that capture both exactness and semantic fidelity.

Figure 1: Example of Visual Question Answer on ABO Dataset

2 Related Work

Visual Question Answering (VQA) is a multimodal task that requires systems to generate natural language answers based on visual inputs. Agrawal et al. (2015) introduced the VQA task combining images with human-annotated questions and answers, emphasizing the open-ended nature of the task and proposing evaluation metrics that account for variability in human responses [1].

While exact match accuracy remains a common metric for evaluating VQA systems, it can be overly rigid for open-ended questions, where semantically correct answers may vary lexically. To address this limitation, the field of Semantic Textual Similarity (STS) offers alternative evaluation strategies. Notably, the SemEval-2012 and SemEval-2013 shared tasks [2, 3] laid the groundwork for comparing textual meaning using sentence-level similarity scores. These efforts have influenced modern metrics such as BERTScore and BGE Score, which leverage contextual embeddings to quantify semantic alignment between generated and reference answers.

Parameter-efficient fine-tuning (PEFT) techniques aim to adapt large pretrained models by updating only a small subset of parameters. Low-Rank Adaptation (LoRA) achieves this by injecting trainable low-rank matrices into existing model weights while keeping the original parameters frozen [4]. This approach significantly reduces the number of trainable parameters required for downstream tasks.

3 Methodology

3.1 Dataset

Amazon Berkeley Objects [5] is a collection of product listings with multilingual metadata, catalog imagery, high-quality 3d models with materials and parts, and benchmarks derived from that data. For this project`abo-images-small` subset was used, which comprises 3GB of images with a maximum resolution of 256×256 pixels.

3.2 Data Curation

Prepare CSV Before Invoking Calls to the VLMs

The dataset named "abo small images" contains the image files, while the corresponding metadata is available in the "abo listing" folder in the form of JSON files. These JSON files were first converted into CSV format by extracting relevant keys—specifically the English-language descriptions (e.g., "en-us", "en-uk"). Only English descriptions were retained. The first level dataset is ready.

At the second stage of CSV preparation, each image is encoded in Base64 format and paired with a prompt that includes its English description. These prompt-image pairs are stored in a separate CSV file.

In the final stage, both the Base64-encoded image and its corresponding prompt are fed into the Vision-Language Model (VLM). The VLM returns 10 diverse question-answer pairs that capture various characteristics of the object(s) in the image. The returned data is then processed to generate a new CSV file that includes the image path, image ID, questions, and answers. This final CSV serves as the dataset for fine-tuning the model.

Visual Question Answering Instructions

You are a **Visual Question Answering Assistant**.

Given an image and its description, your task is to generate **10 diverse visual-only questions** with **single-word answers**. Only ask questions that can be answered *purely by looking at the image*, without needing extra knowledge.

Relevant Topics:

- Object
- Color
- Position
- Count
- Action
- Size
- Shape
- Emotion
- Scene
- Weather
- Material
- Text
- Relation

- Perspective
- Activity
- Place

If the object is a **device**, include questions about its configuration or components.

Format for Question Answer:

```
Q1: What is the shape of the object?
A1: answer
Q2: How many objects are there in the image?
A2: answer
Q3: What is the color of the object?
A3: answer
...
```

Use the given **Image Description** effectively to generate meaningful, image-relevant questions and answers. Do not repeat fixed questions for every image — make them logically relevant to the image's content.

Image Description: *[Insert image description here]*

3.3 Model Selection

Selecting appropriate models for the Visual Question Answering (VQA) task required careful consideration of several key factors. The chosen models needed to be inherently multimodal, capable of processing both visual and textual inputs. Furthermore, they were expected to have strong performance on established VQA benchmarks, while also maintaining a relatively compact parameter footprint suitable for training within the computational constraints of Kaggle GPUs. Additional considerations included training time and the ease of integration into our development environment.

Several state of the art vision language models were evaluated as potential candidates, including CLIP, BLIP, LLaVA, MiniCPM-V-2, LLaMA3-V, mPLUG-Owl3, and Qwen2-VL. After an initial screening based on the criteria mentioned above, we selected **BLIP** and **Qwen2-VL** as our baseline models. Both models are pretrained on large-scale vision-language datasets, are accessible in open-source form, and offer a favorable trade-off between performance and computational requirements.

3.3.1 BLIP-Base

BLIP-Base [8] is smaller compared to its larger counterparts (e.g., BLIP-Large), making it easier to fine-tune on limited compute.Its reduced size allows for quicker experimentation and deployment without sacrificing too much performance.

It is pretrained on large-scale image-text datasets (COCO, VG) using both image-grounded text generation and vision-language matching objectives.Even without fine-tuning, BLIP-Base performs reasonably well on VQA tasks, offering a strong starting baseline.

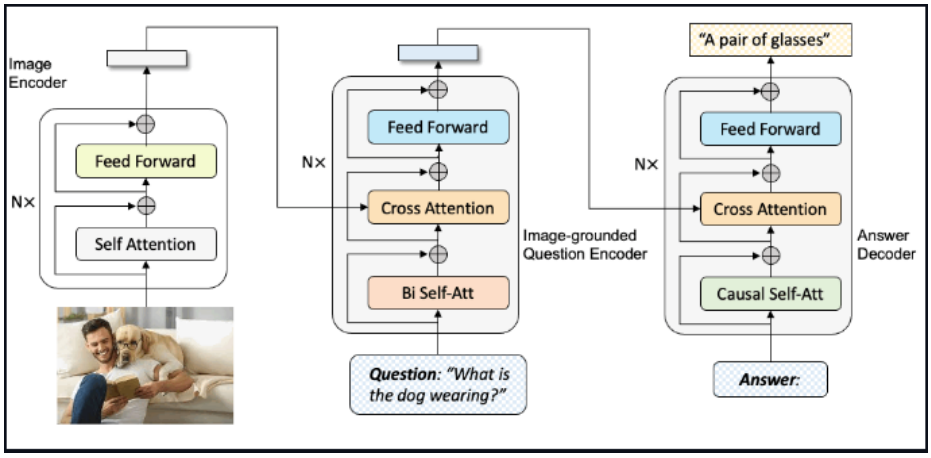

Figure 2: BLIP Architecture

3.3.2 Qwen2-VL-2B-Instruct

Qwen2-VL-2B-Instruct [6,7] is a Large Vision-Language Model (LVLM) it supports a wide range of capabilities, including multilingual image-text understanding, mathematical and code-based reasoning, video analysis, and interactive applications .

Architecturally, it integrates a 675M-parameter Vision Transformer (ViT) encoder with a 1.5B-parameter large language model (LLM) decoder, amounting to a total of 2.21 billion parameters

This makes it a compact yet powerful model, suitable for fine-tuning within constrained compute environments while maintaining high performance across vision-language tasks.

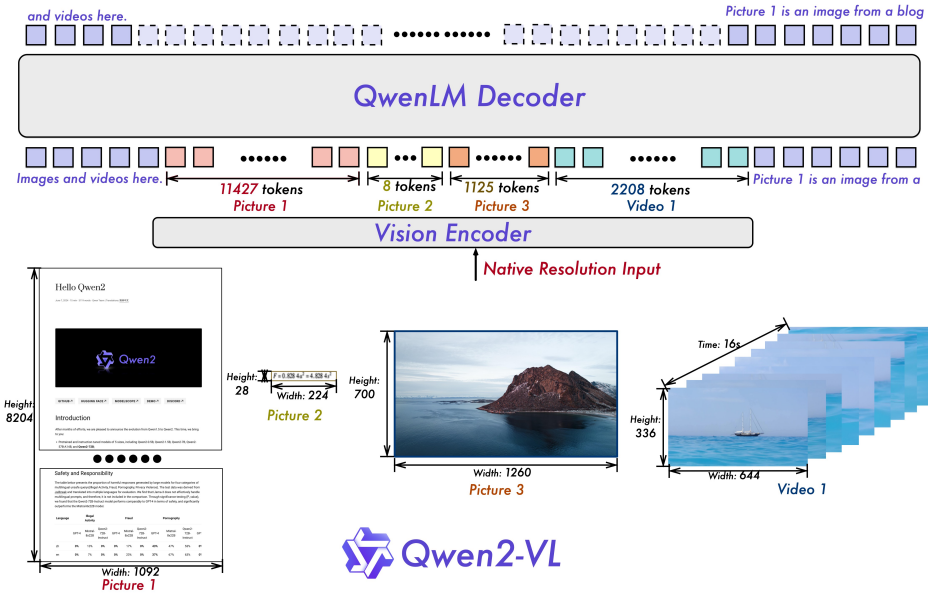

Figure 3: Qwen2-VL Architecture

3.4 Baseline Evaluation

Objective

The models were evaluated on their ability to generate:

- **One-word answers** – when explicitly prompted.
- **Multi-word answers** – when no constraint was provided.

Prompt Structure

- **BLIP-Base VQA**
  - {`question`}
- **Qwen2-VL-2B**
  - **Prompt 1 (One-word)**: {`question`}. `Answer the question in one word only.`
  - **Prompt 2 (Free-form)**: {`question`}.
- **BLIP2 (OPT 2.7B)**
  - **Prompt 1 (One-word)**: `Question:` {`question`}. `Answer in one word only. Answer:`
  - **Prompt 2 (Free-form)**: `Question:` {`question`}. `Give a complete and accurate answer. Answer:`

Despite prompt constraints, BLIP2 and Qwen occasionally produced multi-word responses. In contrast, BLIP-Base VQA reliably produced concise single-word answers even without explicit instructions.

Evaluation Metrics

- **BAAI Similarity**:
  - Word embeddings (e.g., RoBERTa-large) used to compute cosine similarity.
  - Captures semantic closeness (e.g., "cat" vs. "kitten") rather than strict match.
  - Used to calculate Precision, Recall, and F1.
- **BERTScore**:
  - Sentence tokens converted to embeddings, similarity matrix computed.
  - More robust for multi-word or phrase-level matches.
- **Exact Match**:
  - Lowercased, alphanumeric-only answers compared.
  - Penalizes answers with more than one word.

Dataset Optimization

To reduce computational load, the original 100k validation samples were trimmed to 50k. Optimization techniques:

- Grouped by `image_id` to reuse image loading across multiple questions.
- Each image had 10 associated questions.
- Evaluation was done in parallel batches of 5,000 samples and merged for final scoring.

Performance Summary

BLIP2 (OPT 2.7B):

- **BAAI** – Short: 0.69, Long: 0.61, Exact Match: 19.17%
- **BERT** – Short: P/R/F1 = 0.94/0.94/0.94, Long: P/R/F1 = 0.83/0.86/0.84

Qwen2-VL (2B):

- **BAAI** – Short: 0.77, Long: 0.60, Exact Match: 34.16%
- **BERT** – Short: P/R/F1 = 0.96/0.95/0.95, Long: P/R/F1 = 0.80/0.83/0.82

BLIP-Base VQA:

- **BAAI** – Avg Similarity: 0.78, Exact Match (Short): 34.46%
- **BERT** – Short: P/R/F1 = 0.97/0.97/0.97

3.5 Fine-Tuning with LoRA

3.5.1 BLIP Finetuning

We performed supervised fine-tuning (SFT) on the `BLIP-VQA Base` model using a custom VQA dataset.

The loss function was categorical cross-entropy between the predicted and ground truth answers. All samples were preprocessed using the `BlipProcessor`, which tokenized the question, encoded the answer, and transformed the image into pixel values suitable for the ViT backbone.

To enable parameter-efficient fine-tuning, we utilized Low-Rank Adaptation (LoRA). To minimize memory consumption and speed up training, we adopted mixed-precision training (`fp16`). This allowed efficient usage of limited GPU memory without compromising performance.

Training was conducted on Kaggle's NVIDIA Tesla P100 GPU. The model was trained with a batch size of 128. Input padding and truncation were handled automatically by the processor.

Each training sample included a question, an answer, and an image loaded from disk. All training and attention masks were generated and properly aligned with the encoder and decoder components of the model. The model was fine-tuned for 7 epochs on a curated dataset consisting of 100,000 training samples and a 10,000-sample validation set.

3.5.2 Qwen2-VL Finetuning

We performed supervised fine-tuning (SFT) on the `Qwen2-VL` model. The loss function used was categorical cross-entropy over the predicted and actual answers.

To enable parameter-efficient fine-tuning, we utilized Low-Rank Adaptation (LoRA), which reduced the number of trainable parameters from 2.21 billion to approximately 1.7 million. All original model weights were frozen, and only the LoRA-injected adapters were updated during training.Specifically, LoRA was applied to the `q_proj`, `v_proj`, and `o_proj` attention projection modules.

We also adopted mixed-precision training (fp16) to reduce GPU memory consumption and accelerate both training and inference.

Training was conducted on Kaggle's dual NVIDIA T4 GPU setup, with a batch size of 8 per device and a learning rate of `5e-5`. The model was fine-tuned for 10 epochs on a curated dataset consisting of 100,000 training samples and a 1,000-sample validation set.

3.6 Evaluation Metrics

We use both semantic and exact-match based metrics to evaluate the model's output compared to ground truth answers. These metrics are computed using a pairwise comparison between predicted and reference answers, often by constructing a similarity matrix for batch evaluation.

- **BAAI Embedding Similarity:** We compute sentence embeddings (using BAAI models like 'bge') for both predicted and ground truth answers. A square similarity matrix S is formed where:

  $$S_{i,j} = \cos(\vec{y}^{(i)}_{\text{true}}, \vec{y}^{(j)}_{\text{pred}})$$

  We then extract the diagonal values $S_{i,i}$, assuming one-to-one alignment, and average them:

  $$\text{BAAI Similarity} = \frac{1}{N} \sum_{i=1}^{N} S_{i,i}$$

- **Exact Match (Short):** For each example, we compare the predicted and ground truth answers for an exact string match (after normalization such as lowercasing and whitespace stripping). The metric is:

  $$\text{Exact Match \%} = \frac{\text{\# of exact matches}}{N} \times 100$$

- **BERTScore:** This metric evaluates semantic similarity using contextual embeddings from BERT. We construct a precision, recall, and F1 score for each sample:

  $$\text{Precision}_i = \frac{|\text{matched tokens in prediction}_i|}{|\text{tokens in prediction}_i|}$$

  $$\text{Recall}_i = \frac{|\text{matched tokens in prediction}_i|}{|\text{tokens in reference}_i|}$$

  $$\text{F1}_i = 2 \times \frac{\text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$$

  The final metric is the average across all N samples:

  $$\text{Avg. BERTScore (F1)} = \frac{1}{N} \sum_{i=1}^{N} \text{F1}_i$$

We run these metrics on both pre-trained models and after fine-tuning, comparing performance improvements across the board.

4 Results

Table 1: Performance of BLIP VQA Before and After Fine-Tuning

Metric	**Before FT**	**After FT**
BAAI Similarity	0.78	0.86
Exact Match (Short)	34.46%	60.60%
BERTScore Precision (Short)	0.97	0.99
BERTScore Recall (Short)	0.97	0.99
BERTScore F1 (Short)	0.97	0.99

Table 2: Performance of Qwen Before and After Fine-Tuning

Metric	**Before FT**	**After FT**
BAAI Similarity	0.77	0.87
Exact Match (Short)	34.16%	62.76%
BERTScore Precision (Short)	0.96	0.99
BERTScore Recall (Short)	0.95	0.99
BERTScore F1 (Short)	0.95	0.99

References

[1] Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh, *VQA: Visual Question Answering*, Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2425–2433.

[2] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre, *SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity*, Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM), 2012, pp. 385–393.

[3] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo, **SEM 2013 Shared Task: Semantic Textual Similarity*, Second Joint Conference on Lexical and Computational Semantics (*SEM), 2013, pp. 32–43.

[4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen, *LoRA: Low-Rank Adaptation of Large Language Models*, arXiv preprint arXiv:2106.09685, 2021.

[5] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F. Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik,
ABO: Dataset and Benchmarks for Real-World 3D Object Understanding,
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[6] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. *Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution.* arXiv preprint arXiv:2409.12191, 2024.

[7] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. *Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond*. arXiv preprint arXiv:2308.12966, 2023.

[8] Junnan Li, Dongxu Li, Caiming Xiong, Steven Hoi *BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation* https://arxiv.org/abs/2201.12086